0010

ESD-TR-73-51, Vol. I

COMPUTER SECURITY TECHNOLOGY PLANNING STUDY

James P. Anderson

October 1972

DEPUTY FOR COMMAND AND MANAGEMENT SYSTEMS
HQ ELECTRONIC SYSTEMS DIVISION (AFSC)
L. G. Hanscom Field, Bedford, Massachusetts 01730

Approved for public release;
distribution unlimited.

(Prepared under Contract No. F19628-72-C-0198 by James P. Anderson & Co., Box 42, Fort Washington, Pa. 19034.)

LEGAL NOTICE

When U.S. Government drawings, specifications or other data are used for any purpose other than a definitely related government procurement operation, the government thereby incurs no responsibility nor any obligation whatsoever; and the fact that the government may have formulated, furnished, or in any way supplied the said drawings, specifications, or other data is not to be regarded by implication or otherwise as in any manner licensing the holder or any other person or conveying any rights or permission to manufacture, use, or sell any patented invention that may in any way be related thereto.

OTHER NOTICES

Do not return this copy. Retain or destroy.

LETTER OF TRANSMITTAL

Major Roger Schell, (MCI)
Air Force Systems Command, USAF
L. G. Hanscom Field
Bedford, Mass. 01730

Dear Major Schell:

This is a summary of our work as the Computer Security Technology Planning Study Panel. Each member of the panel has participated, and signs as an individual contributor, not as a representative of his or her organization.

Although there may be some disagreement over specific points, the panel totally concurs with overall substance and interest of the development program. We strongly recommend that the Air Force initiate such a program without delay.

Very truly yours,

E. L. Glaser, Chairman
Computer Security Planning Study

James P. Anderson

Eldred Nelson

Melvin Conway

Bruce Peters

Daniel J. Edwards

Charles Rose

Hilda Faust

Clark Weissman

Steven Lipner

COMPUTER SECURITY TECHNOLOGY PLANNING STUDY
VOLUME I – EXECUTIVE SUMMARY

JAMES P. ANDERSON

FOREWORD

This is Volume I of a two volume report of the Computer Security Technology Planning Study conducted during the period 2 February – 1 September 1972 by James P. Anderson & Co., in support of Project 6917 under contract F19628-72-C-0198. The study was conducted using a panel of recognized authorities in the field of computer security. This report is an integration of the individual contributions of the panel members listed below.

Professor E. L. Glaser, Case Western Reserve University, Chairman

J. P. Anderson, James P. Anderson & Co.

Dr. Melvin Conway, Private Consultant

Mr. Daniel J. Edwards, National Security Agency

Miss Hilda Faust, National Security Agency

Mr. Steven Lipner, The MITRE Corporation

Dr. Eldred Nelson, TRW, Incorporated

Mr. Bruce Peters, System Development Corporation

Dr. Charles Rose, Case Western Reserve University

Mr. Clark Weissman, System Development Corporation

It must be emphasized that the views and recommendations contained in this report represent the independent and individual views of the participants, and in no way represents official views of their organizations.

The contributions and encouragement of the program manager, Major Roger Schell, USAF (ESD/MCI) are gratefully acknowledged.

REVIEW AND APPROVAL

This technical report has been reviewed and approved.

MELVIN B. EMMONS, Colonel, USAF
Director, Information Systems Technology
Deputy for Command and Management Systems

ABSTRACT

The results of a planning study for USAF multilevel computer security requirements are presented. The study recommends research and development urgently needed to provide secure information processing systems for command and control and support systems for the Air Force

PREFACE

This study was conducted by a panel of authorities from university, industrial, and Government organizations. It addresses the problems uncovered by an independent working group of working level staff officers in Air Force commands using computers. The panel met as a body six times, and conducted independent study between panel meetings.

The principal unsolved technical problem found by the working group was that of how to provide multilevel resource and information sharing systems secure against the threat from a malicious user. This problem is neither hopeless nor solved. It is, however, perfectly clear to the panel that solutions to the problem will not occur spontaneously, nor will they come from the various well-intentioned attempts to provide security as an add-on to existing systems.

The reason that an add-on approach, which looks so appealing, will not suffice is that in order to provide defense against a malicious user, one must design the security controls into the operating system of a machine so as to not only control the actions of each user, but of the many parts of the operating system itself when it is acting on a user's behalf. It is this latter requirement that invalidates the concept of providing only those controls required by the security level of the information being processed on a system. The issue of computer security is one of completeness rather than degree, and a complete system will provide all of the controls necessary for a mixture of all security levels on a single system. It is the notion of completeness that compels one to take the position that security must be designed into systems at their inception.

The approach recommended in the development plan is to start with a statement of an ideal system, a model, and to refine and move the statement through various levels of design into the mechanisms that implement the model system. Other elements of the plan address ancillary developments needed to reduce costs or to support common applications.

The plan described in this report represents a coherent approach to attacking these problems. It is our opinion that attempting to solve the problem by piecemeal application of parts of this plan will not produce the desired results.

TABLE OF CONTENTS

Section		Page
I	MANAGEMENT SUMMARY	1
	1. 1 USAF Computer Security Problem	1
	1. 2 Concept of Malicious Threat	1
	1. 3 Penetration of Systems	2
	1. 4 Inadequacy of Contemporary System Designs	2
	1. 5 Why This Study?	3
	1. 6 Operational and Economic Considerations	4
	1. 7 Overview of the Management Summary	5
II	REQUIREMENTS	6
	2. 1 Source of Requirements	6
	2. 2 Type of Systems	6
	2. 3 Requirements Trends and Security Problems	7
III	ELEMENTS OF A SECURE SYSTEM MODEL	8
	3. 1 Emerging Design Principles	8
	3. 2 Outline of a Model	8
	3. 2. 1 Basic Concepts	8
	3. 2. 2 Operating Requirements	9
	3. 2. 3 The Appeal of This Approach	11
	3. 3 Reference Validation Mechanism Design	11
	3. 4 Building a Secure System	11
	3. 5 Understanding Limitations of Contemporary Systems	12
	3. 6 Applicability to Operational Problems	13
	3. 7 What The Principles Do Not Cover	13
IV	THE DEVELOPMENT PLAN	14
	4. 1 Advanced Development Plan — Secure Open-Use Systems	14
	4. 2 Supporting Engineering Developments	16
	4. 3 Related Advanced Development Plan — Developments for Interim Solutions to Current Problems	18
	4. 3. 1 Secure Limited Use Systems	19
	4. 3. 2 Repair of Current Systems	21
	4. 4 Exploratory Development Plan	23
	4. 5 Summary of The Development Plan	24
V	ECONOMIC CONSEQUENCE OF INADEQUACIES OF CURRENT SYSTEMS	28
	5. 1 Cost of Inadequate Systems	28
	5. 2 Cost to Penetrate Systems	29
VI	CONCLUSIONS	32

SECTION I

MANAGEMENT SUMMARY

1. BACKGROUND

1.1 USAF Computer Security Problem

This is the final report of the USAF Computer Security Technology Planning Study Panel. The Panel was charged to develop a comprehensive Research and Development Plan leading to the satisfaction of the requirements for open use, multi-user, resource shared computer systems which process various levels of classified and unclassified information simultaneously from terminals in both secure and unsecure areas. The major computer security problem of the USAF stems from the fact that there is an urgent requirement to provide shared use of computer systems among a user population not uniformly cleared both for reasons of operational need and economy. Presently available systems are unable to provide the level of protection needed for this kind of use and the designs are not certifiable. This report presents a research and development plan to guide the work leading to the achievement of secure multilevel computer systems for the Air Force.

1.2 Concept of Malicious Threat

The malicious user concept arises from the requirements for open use systems. Present day computer systems are largely closed use systems; that is, systems securing a homogeneously cleared user population. The major threat to these systems is that of external penetration. The external penetration threat is countered by using combinations of physical, procedural and communications security techniques. These techniques, some highly advanced, are the bulk of the present state-of-the-art in computer security. In effect, the defense against external penetration surrounds the system and its user community with a barrier that must be breached before the system can be compromised. By adopting a uniform clearance (to the highest level of information contained in the systems), the threat of internal penetration is eliminated by definition.

The requirements working group supporting this study identified a number of operational requirements not currently being met with existing commercially available hardware and systems. The most significant of these were the growing economic and operational pressure for online multilevel secure operations, and for open use secure systems. These two requirements are both concerned with the same issue(s) – that of providing adequate protection to classified information in systems where all users are <u>not</u> cleared for all of the information contained in the system.

In the case of open use systems there is an implication of unprotected communications lines for subscribers not performing classified processing that increases the exposure to external penetration. In multilevel systems, it is often assumed that all communications are protected; while this minimizes or eliminates the threat of external

penetration, multilevel systems by definition leaves the threat of a malicious user unchecked. By the term multilevel we mean to include the concept of uncleared. users as well as users cleared for Confidential, Secret, and Top Secret (or any subset of these) all sharing a single system. The significant aspect of open use multilevel systems is that control over the user population implied by the homogeneous clearances required in closed systems no longer exists. Without such controls there exists a threat that penetration of the system will be attempted by a malicious user.

The technical issue of multilevel computer security is concerned with the concept of malicious threat. By this we recognize that the nature of shared use multilevel computer systems present to a malicious user a unique opportunity for attempting to subvert through programming the mechanism upon which security depends (i. e., the control of the computer vested in the operating system). This threat, coupled with the concentration of the application (data, control system, etc.) in one place (the computer system) makes computers a uniquely attractive target for malicious (hostile) action. Recognition of the implication of malicious threat is important to understanding the security limitations surrounding application of contemporary computer systems. The threat that a single user of a system operating as a hostile agent can simply modify an operating system to by-pass or suspend security controls, and the fact that the operating system controlling the computer application(s) is developed outside of USAF control, contribute strongly to the reluctance to certify (i. e., be convinced) that contemporary systems are secure or even can be secured.

While we emphasize the threat from a malicious user, we are not unmindful of other security threats and risks. The problems of accidental spillage of classified information, physical penetration of system sites, interference with or intercept of communications, mishandling of classified material and the like are serious. They require attention in the design, implementation and operation of a system. To a large extent, these problems are common to any information system processing classified information and can be solved by well understood techniques. However, the malicious user in the context of a resource shared system presents a new type of threat, control of which is necessary before the objective of full use of shared computer systems can be realized.

In order for a system to be suitable for open use multilevel operations, it must be conclusively demonstrated that a malicious user cannot gain control of the system or unauthorized access to data in the system. In general, it must be possible for the malicious user to be able to promote and execute arbitrary programs in order to seize control of a system. However, even with more restricted capabilities, he may still be able to gain unauthorized access to data in the system if the system was designed or implemented improperly, or trapdoors have been placed in the operating system.

1. 3 Inadequacy of Contemporary System Designs

The reason that it is difficult to provide technical security in contemporary computer systems is that the technical foundation (i. e., design) of the hardware and software is totally inadequate to withstand malicious attack. This is because the

designers never considered other than a benign environment where seeming violations of the system are presumed to be accidental. In this milieu, the idea that anyone would deliberately attempt to seize control of a system or penetrate it was dismissed as unrealistic for the commerical world where most of the customer base is found. As a consequence, the controls that do exist are there to contain accidents or errors.

Unless security is designed into a system from its inception, there is little chance that it can be made secure by retrofit. Contemporary operating systems such as that used for WWMCCS are not built in accordance with a unified set of principles of computer security and there is no way to make local patches to the systems to compensate for this.

A large part of the design problem is attributable to the absence of models as a medium for translating security requirements to technical specifications and as a source of acceptance criteria for evaluating the product. Without such models, system developers are forced to apply ad hoc security related techniques throughout the design and implementation of the system. This approach inevitably leads to exploitable flaws, and makes the security assessments necessary for certification virtually impossible. Because the techniques are ad hoc, there is no way to determine whether or not all portions of the operating system requiring security controls have the appropriate techniques applied. Neither is it possible to determine where in the operating system design such controls are required. Lacking a set of principles adhered to strongly in the design of a system, one finds that there is no way to determine when a secure system has been achieved through any of the presently known testing methods. Under these circumstances, it is little wonder that contemporary systems cannot be certified.

1.4 Penetration of Systems

There is little question that contemporary commercially available systems do not provide an adequate defense against malicious threat. Most of these systems are known to have serious design and implementation flaws that can be exploited by individuals with programming access to the system. As an instance of this, we note that the Honeywell 6000 Series operating system has a number of major flaws that would permit a user programmer to subvert the nominal security controls that exist in the system. The design and implementation flaws in most contemporary systems permit a penetrating programmer to seize unauthorized control of the system, and thus have access to any of the information on the system. While the techniques for achieving this access on contemporary systems vary (see Appendix I in Vol. II) they ultimately boil down to gaining, either directly or indirectly, an unauthorized access capability to classified data.

1.5 Why This Study?

The reason for the study is that there is virtually nothing now being done that is applicable to the problem of secure computing in the USAF. Although the problem of computer security is recognized in the recently completed CCIP-85 study, the

development program recommended as a consequence of that work is not yet funded, and addresses only certain of the problems. What work is being done in this area at present is sporadic and uncoordinated. It is the belief of the panel that the problem requires a comprehensive and coordinated attack.

Previous work in computer security has been mostly concerned with adapting existing manufacturer supplied hardware and software systems in completely closed environments and adding to them automated versions of external procedural controls. There has also been some related activity in the form of 'tiger teams' that have expended a moderate amount of energy in demonstrating the security inadequacy of both standard commercial systems and those ostensibly modified to provide security controls. The value of 'tiger teams' in testing computer security is questionable because the results of the effort are highly dependent on the quality and experience of the personnel assigned to the teams. Even if corrections are made as a result of flaws found by a team, there is no assurance that all flaws have been found and corrected. The activities of the tiger team can only reveal system flaws and provide no basis for asserting that a system is secure in the event their efforts are unsuccessful. In the latter event, the only thing that can be stated is that the security state of the system is unknown. It is a commentary on contemporary systems that none of the known tiger team efforts has failed to date.

The study conducted by the Defense Science Board's Task Force on Computer Security (RAND Report R-609), while an important milestone, did not have the impact intended, and may have had a negative effect due to its specification of necessary, but not sufficient, criteria for evaluating hardware and software suitable for secure operations. More recent efforts, such as those supported by the Advanced Research Projects Agency (ARPA) appear to be focusing on one or more interesting (to the principal investigator) research problems, but do not evidence a comprehensive or cohesive approach to solve the USAF Computer Security Problem.

1.6 Operational and Economic Considerations

The consequences of the inadequate security mechanisms in current Air Force computer systems are both the potential for loss of information critical to national security by enemy penetration and a higher cost of operation. Operational requirements for multilevel systems are based on the need for rapid access to and dynamic sharing of information at varying levels of classification. At present, these requirements cannot be met without great risk of penetration and compromise. Higher costs of operation include costs due to separate computers for separate applications, restricting use of remote terminals, costs of physical protection of remote terminals and associated crypto devices, and the costs of clearing all user personnel to the highest level of classified information processed by a system. Pursuing the plan recommended in this report will have a significant effect in reducing these costs — perhaps yielding a reduction of 20 to 40 percent the cost of operating USAF computer systems that handle classified data.

1.7 Overview of the Management Summary

Section 2 summarizes the Air Force requirements that are impacted by the problem of computer security. Section 3 presents a brief background summary of the technical approach recommended for the advanced development plan to achieve secure multi-user open-use systems. The development program is given in Section 4. Section 4.1 summarizes the advanced development program to achieve open-use security. Supporting engineering development is summarized in Section 4.2. An advanced development program to provide interim solutions to security problems on current systems is summarized in Section 4.3. This program is based on expected early results from the open-use system development. An exploratory development program complementing the advanced development program is summarized in Section 4.4. A cost of and schedule summary of the entire development program is contained in Section 4.5. Section 5 discusses some of the cost consequences of attempting to continue with present equipment and no developments in this area. Section 6 is a summary of conclusions.

SECTION II

REQUIREMENTS

2.1 Source of Requirements

The operational requirements used to motivate the panel activities were derived by a working group whose objective was to identify the directions in which USAF computer use was moving and the relation of those directions to computer security. The requirements working group was composed of working level staff officers of Air Force commands that are computer users. These officers presented descriptions of their commands existing and planned computer usage, and the computer security problem as perceived by their commands. The Air Force commands that participated in this work were:

Air Force Logistics Command	(AFLC)
Air Force Data Services Center	(AFDSC)
Satellite Control Facility	(SAMSO)
NORAD/Aerospace Defense Command	(NORAD)
Air Force Communications Service	(AFCS)
Air Force Global Weather Center	(AFGWC)
Strategic Air Command	(SAC)
Air Force Security Service	(AFSS)
Military Air Lift Command	(MAC)
Electronic Compatibility Analysis Center	(ECAC)

2.2 Type of Systems

The systems examined by the working group spanned a broad range of functions from systems that support general-use programming in both batch and time-sharing modes (such as the USAF Data Services Center), to systems that perform only dedicated prespecified functions such as responding to user queries or (like AUTODIN) acting as message switch systems. The bulk of the systems examined were query and transaction processing systems such as Advanced Logistics System or the Military Airlift Command Information Management System (MACIMS) which provide query and transaction processing to many online users and are supported by a general programming and software maintenance staff. Most of the computer systems examined by the requirements group were medium and large scale, because these are the primary kinds of systems with the capability to make resource sharing economically feasible for the applications contemplated. Virtually all manufacturers' equipments were represented.

2.3 Requirements Trends and Security Problems

The primary security related operational requirements noted by the users were:

a. Online Multilevel Secure Operation (AFLC, AFDSC, NORAD, AFGWC, SAC, MAC, ECAC)

b. Open Operation (AFDSC, MAC, ECAC, AFGWC)

c. Transaction Systems (AFLC, MAC)

d. General Programming (AFLC, AFDSC, NORAD, AFGWC, SAC, MAC, ECAC)

e. Networks (all)

In addition to the requirements noted above, there were a number of problems which users had perceived as paramount for their current operations; the lack of adequate computer security support found in contemporary systems, difficulties in providing secure operation by ad hoc additions to the equipment/software base, terminal security, and media (e.g. magnetic disc, drum, ontape memory) security and media declassification.

The requirements, presented roughly in the order of their importance indicated by the users, can only partially be met by present technology. By using cleared personnel throughout the development process, present technology provides a potentially suitable base for realization of technically secure online multilevel computing environments only for limited applications. For example, it is feasible to provide a multilevel secure transaction or query system on present equipment provided there is no other use of the system. Similarly, multilevel secure dedicated message switching systems like AUTODIN can also be realized.

As the requirements for more general use intrude (including programming development on an operational system, general programming use and the like), present systems are unable to provide protection against malicious users. Thus we find it is not now possible to provide certifiably secure multilevel systems (either online or batch) where general programming use is involved.

The security condition of networks is even less structured than that of most applications. Computer networks that have one or more nodes that can be accessed by users with clearances below the highest level of information in the network, constitute multilevel networks. The security threat posed by such operations is that, in general, the computer to computer communications are accepted as valid on the questionable basis that the other computer has a high security reliability. However, if control of a node can be exercised by a malicious users, the entire network may be compromised. While there are growing requirements for interconnecting computer systems into networks and several networks (Air Weather Network, 465L SACCS, BUIC, and AUTODIN) already exist, the dimensions of the security problem are unknown. More information is needed on both the networks and their security requirements. For this reason we are recommending that network security be included in the exploratory development program.

SECTION III

ELEMENTS OF A SECURE SYSTEM MODEL

3.1 Requirements For Defense Against a Malicious User

Until now, the principal threat has been seen to be an external penetration. The primary defense against external penetration has been that of preventing access to any part of the system or its data. The malicious user concept on the other hand has bypassed this form of defense by assuming that the malicious user has legitimate access to a system. Taken in the context of open use systems with general programming available to all users, it is clear that the defense against a malicious user must reside in the process that controls the operation and execution of arbitrary programs. The principal requirement is that of being able to precisely control and limit the references a program can make to other programs and data to just those authorized (by an external authority) for the user on whose behalf the program is executed.

3.2 Outline of a Model

The panel believes that the principles described below provide the most promising approach to achieving systems secure against the threat of malicious users. It is anticipated that these principles, can be applied during the design of an operating system to isolate those elements concerned with defense against internal penetration. These elements must then be explicitly required to follow the constraints on access to information specified by the military security system. The collection of system elements constrained by the security rules form a high-level model for a secure operating system. This model can be used as a basis for the detailed design of effective and certifiable hardware and software access control[1] and security mechanisms. Such a model will provide a complete description of the security components of an operating environment, along with a set of primitive operations on the representation of these elements.

3.2.1 Basic Concepts

The basic concept upon which multilevel secure computing systems can be based is that of controlled sharing. Explicit control must be established over each user's (programs) access to any system resource which is shared with any other user or (system) program. Essential to this concept is the requirement that each subject of the system (viz. system entities such as a user or a program which can access system resources) and each object (viz. system entities such as data, programs, peripheral devices, main memory and subjects which can be accessed by other subjects) must be identified and interrelated according to their authorized accessibility.

---

[1] Access control mechanism — a combination of procedural, hardware and software checks that validate a user's right to make use of a computer system.

One of the most promising developments of this idea is the concept of a reference monitor[2] which enforces the authorized access relationships between subjects and objects of a system. This is illustrated in Figure 1. An implementation of the reference monitor concept is a reference validation mechanism[3] that validates each reference to data or programs by any user (program) against a list of authorized types of reference for that user. The authorized access relationships between subjects and objects are defined in terms of privilege (e. g. READ or WRITE applied to data objects and EXECUTE applied to program objects). As a means of depicting these relationships, it is convenient to use a matrix, with subjects making access listed opposite rows and objects to which access is made listed above the columns. The entries in the matrix define for each subject whether access is permitted to the object and with what privileges. An example of such a matrix is shown in Figure 2.

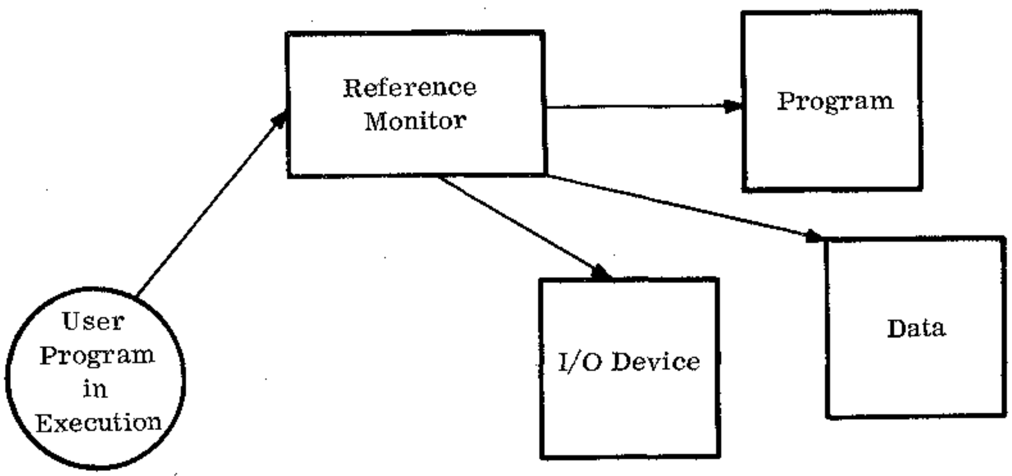


Figure 1. Reference Monitor

3. 2. 2 Design Requirements

Accompanying the concept of Reference Monitor are other essential design requirements. They are:

a. The reference validation mechanism must be tamper proof.

---

[2]Reference monitor concept — the notion that all references by any program to any program, data or device are validated against a list of authorized types of reference based on user and/or program function.

[3]Reference validation mechanism — that combination of hardware and software which implements the reference monitor concept.

Objects / Subjects	Other Subjects			Files			I/O Devices
	S_1	S_2	S_3	F_1	F_2	F_n	$D_1 \ldots D_n$
S_1		BLOCK ENABLE		READ WRITE			
S_2			STOP		UPDATE		
S_3				DELETE	EXECUTE		
⋮							

Figure 2. An Access Matrix[4]

b. The reference validation mechanism must always be invoked.

c. The reference validation mechanism must be small enough to be subject to analysis and tests, the completeness of which can be assured.

Each of these requirements is significant, for without them the mechanism cannot be considered secure. The first is obvious, since if the reference validation mechanism can be tampered with, its validity is destroyed, as is any hope of achieving security through it. The second requirements of always invoking the reference validation mechanism simply states that if the reference validation is (or must be) suspended for some group of programs, then those programs must be considered part of the security apparatus, and be subject to the first and last requirements. The last requirement is equally important. It states that because the reference validation mechanism is the security mechanism in the system, it must be possible to ascertain that it works correctly in all cases and is always invoked. If this cannot be achieved, then there is no way to know that the reference validation takes place correctly in all cases, and therefore there is no basis for certifying a system as secure.

---

[4]From Graham, G. S., and Denning, P. J. "Protection-Principles and Practice", Proceedings SJCC, 1972, pp 417-429.

3.2.3 The Appeal of This Approach

The appeal of the ideas outlined is quite strong. First, as the basis for a security model, they are a simple and easily understood set of principles. Second, there appears to be no restriction as to what a model built on these principles can cover, or to the variety of situations to which it can be applied. It is immaterial whether for a given access matrix the security objects represent files and the subjects represent users authorized to access the files (for Reading or Update), or whether the objects represent programs accessed by subjects that are either users or programs.

3.3 Reference Validation Mechanism Design

It must be obvious that the approach is not of itself a model, although the outlines of such a model are quite clear. What is needed beyond a model is a design for a reference validation mechanism that mirrors the model and is faithful in its exercise of the principles upon which the approach to the model is based. While the design of the reference validation mechanism will not result in an operating system per se, it should help provide a foundation upon which a secure operating system can be developed.

It is at the point of transforming a model into a design that the efficiency of the validation mechanisms becomes important. While a programmed interpreter may be suitable as a reference monitor in a query system, it will be necessary to utilize hardware interpreters in order to provide general use secure systems. There are systems with the appropriate hardware upon which an efficient reference monitor design can be realized.

Since each operational computer system having a security requirement must be certified to have an acceptable secure mode of operation, the model referred to in the preceding paragraphs must be certified to form an acceptable basis for a secure system. This certification requires a proof that the system represented by the model is secure and a demonstration that an implemented instance of the model corresponds to the model. The proof of model security requires a verification that the modeled reference validation mechanism is tamper resistant, is always invoked, and cannot be circumvented.

After the model has been certified, each system developed according to the model must also be certified to have an acceptably secure mode of operation. This includes verification that the system as implemented conforms to the model and does not perform actions that would circumvent the security mechanisms specified by the model.

3.4 Building a Secure System

The approach to obtaining a secure system involves first defining what threats the system is to be secure against, and then defining a conceptual design that can be shown to provide the required protection (i.e. the model). In effect, the model formally defines an ideal system that complies with military security requirements

and provides a basis for testing a subsequent implementation. In the approach outlined above, we have concentrated on the threat posed by a potential malicious user, since it is against this threat that contemporary systems most frequently fail. Having an 'ideal' system in the form of a model, the next step is to obtain an implementation design that provides mechanisms that meet the requirements of the ideal system. The design process may be iterative, with increasing detail provided at each iteration. At each stage of the transformation of the model into a design it is necessary to demonstrate formally that the design remains faithful to the precepts of the model.

Because the major security related issue being faced is how to control a (presumed malicious) user's ability to reference programs or data, those systems (such as Multics or other descriptor-driven systems) that have already dealt with this problem (not entirely for security related reasons) appear to offer the best vehicle for implementing a secure system.

After the design is complete, there remains the vital task of correctly implementing the design on some computer system to provide the nucleus of a secure system. Note that the nucleus includes all the security protection mechanisms that are properly a part of a computer system, not just those necessary to control a users capability to reference programs and data. For some potential applications of such a system, the process of constructing a secure system may require that both the hardware and the operating of system be produced in a security controlled environment. Structured programming and program-proving techniques can be used to assure that the model, design and implementation correspond. The implementors must assure that the environment for producing security control software is itself secure; for example, that compiler and linkage editors are either certified free from "trap-doors", or that their output can be checked and verified independently.

3.5 Understanding Limitations of Contemporary Systems

Even without a fully developed model, the limitations of contemporary systems in achieving secure operations become evident, and for very specific reasons. As an example, consider the general (programming) use of the HIS 635 with GCOS III. The hardware of the HIS 635 provides a kind of reference monitoring in the form of the bounds register that limits a user's program to direct access of his own memory area, and keeps it out of all other memory in the system. However, in order to perform some essential functions, notably I/O, the user program makes use of the GCOS III supervisor. When this program is in operation on behalf of a user program, it is in effect an extension of that user. However, most of GCOS III operates in supervisor state, in which the bounds checking is suspended. Thus we find the HIS 635/GCOS III violating the principle that the reference validation mechanism must always be invoked. If we try to defend this by saying GCOS III is the reference validation mechanism, we find the sheer size of GCOS III including supporting software (200,000 - 300,000 instructions) in violation of the principle that the reference validation mechanism must be small enough to permit analysis and testing, the completeness of which can be assured. The success of a number of security penetration exercises against GCOS III is sufficient comment regarding the self protection principle. When considered

in the context of malicious threat, the inability to certify the HIS 635/GCOS III system for secure general programming operations is evident. A similar criticism can be directed to all other commercially available systems.

3.6 Applicability to Operational Problems

A secure query system (subject to the risk of malicious threat in the construction of the query system, or the hardware/software base upon which it is formed) could be installed on a machine such as the HIS 6000 series computer. This requires that the reference validation mechanism is made part of the query system design, that it properly checks all references to the data base, that the operating principles are strictly followed, and that the users have only the facilities of the query system at their disposal and cannot use them (or the system) to do general programming. The primary risk in such a system is that a 'trap door' could be inserted in the query system or the underlying hardware/software base, and could be activated by a malicious user to suspend the security controls built into the system. The implication of this risk is that all persons involved in the query system design and in the design and maintenance of the operating system base must be cleared and of the highest reliability. This merely illustrates the scope of the security risk attendant to using contemporary software not produced under controlled conditions for building 'secure' systems. Where 'maintenance' and improvements of the software base is left to the manufacturer, as is the case in nearly all contemporary systems, there is no way to certify (guarantee) that one or more 'trap doors' have not been inserted in the operating system. It is for this reason that the development plan calls for the security related functions to be centralized in one or a few program modules produced by fully cleared individuals in a secure environment.

3.7 What The Principles Do Not Cover

The reference monitor concept is directed to overcoming the threat of a malicious user in systems supporting general programming. The concept is predicated upon positive identification (authentication) of all users at all times, the application of adequate physical security measures and procedures to protect the system, file media, terminals etc., and proper protection of the communications between users and the system. These areas are not included in the reference monitor concept. However, specific security measures associated with these areas can be derived from current technology. In addition, the model assumes properly operating or fault tolerant hardware and physical components. While the present state of the art is not sufficient to provide guaranteed fault tolerance, much work is being conducted in this area. Further, experience with a number of large scale resource-sharing systems indicates that this problem generally has little security impact except for a few isolated incidents.

SECTION IV

THE DEVELOPMENT PLAN

This section summarizes the recommended advanced development plan for a secure computer (4. 1), supporting engineering developments (4. 2), a related advanced development plan to provide interim solutions to some of the security problems with contemporary systems (4. 3), and an exploratory development plan to complement the other developments (4. 4). Volume II of the report expands on both the recommendations and rationale for the plans presented.

4. 1 Advanced Development Plan — Secure Open-Use Systems

The recommended approach for achieving the objective of a secure, open-use, multilevel resource sharing system supporting general programming use is shown in Figure 3. Adherence to this approach will result in a prototype of a secure computing system that is designed to efficiently implement the access control mechanisms and security related functions derived from a model of secure computer operations. The model, satisfying the concepts of the preceding section, is the vehicle for identifying and collecting in one place all access control and security related functions of a system and provides the basis for the verification of security related system elements. The access control, reference validation mechanism and security related functions are referred to as the 'Security Kernel'.[5] There are two approaches to be explored. The "shared operating system" approach, the more obvious conception, sees the security kernel at the heart of a single, new, secure operating system that is shared among its users as is current practice (e. g. GECOS, OS/360, etc.). The "shared machine" approach, though less common (e. g. CP-67, VM/370), uses a security kernel as the heart of a control program (i. e. , an operating system for running an operating system) that security shares the physical computer hardware among its users in a fashion that lets each user have a different "virtual machine" operating an operating system of his choice. The key issues will be the generality of a system using a reference validation mechanism, and the efficiency with which the validations can take place. Supporting this development are current computers that use 'descriptors'[6] to define and enforce control of access to programs and data. The descriptor based computers are of particular importance to achieving a secure

---

[5]Security kernel — the software portion of the reference monitor and access control mechanisms.

[6]Descriptor — one or several computer words which define the reference rights of a program to a program, data or device. Descriptors are a possible implementation of a portion of a reference monitor mechanism. Descriptors may be interpreted by hardware or software and are normally not available to the program making the reference.

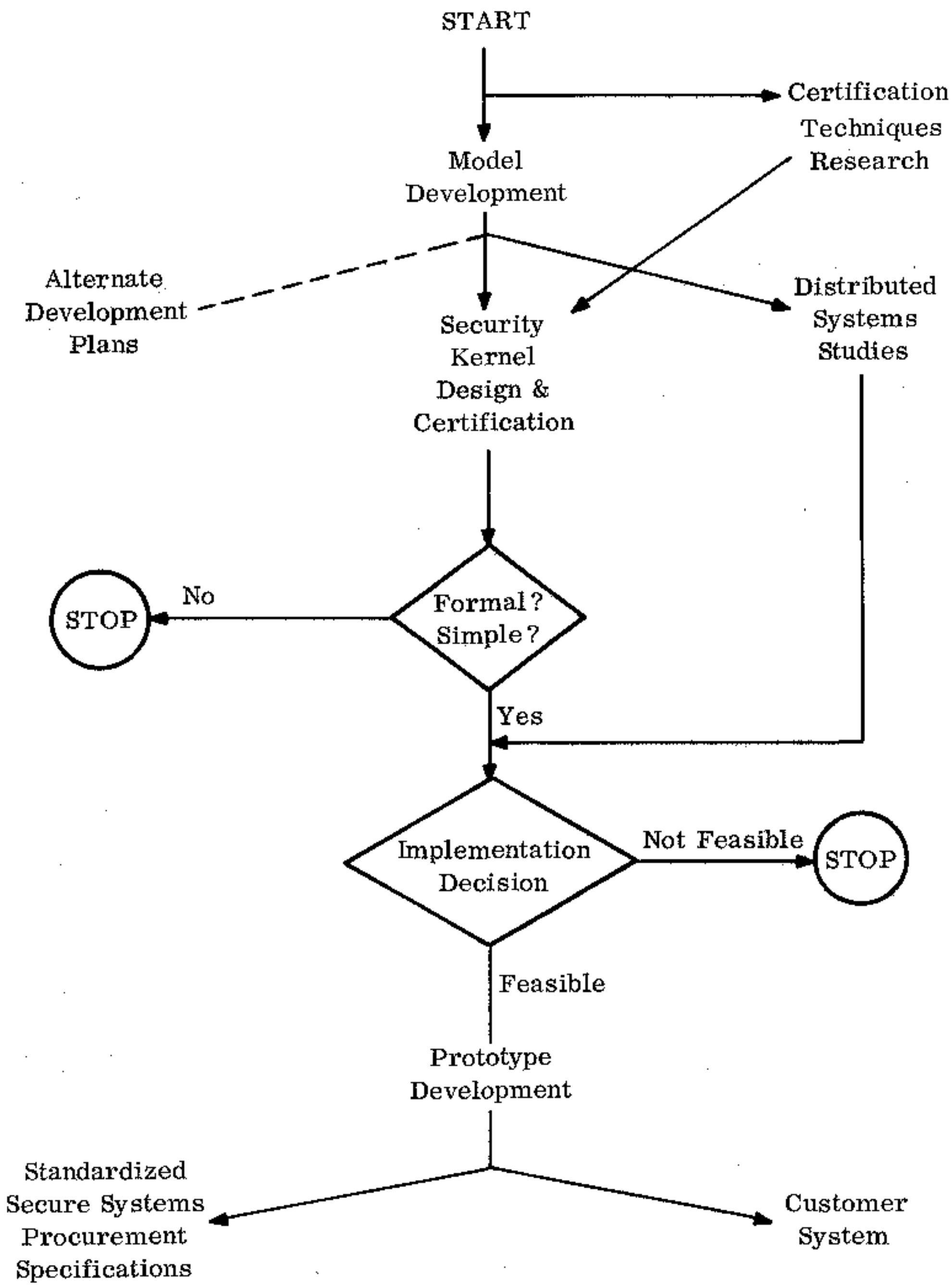


Figure 3. Approach For Secure Open-Use Systems

general programming capability, since they provide a means of efficiently implementing of the kind of reference validation that is central to the concept of the model of secure computing.

Because the model of secure computing is directed to identifying and isolating the essential security features of an operating environment, it is recommended that the alternative of implementing the physical and functional distribution of operating system functions over physically segregated machines, (operating in a multi-computer or multi-processor network), also be evaluated as a candidate system design.

Included in the development of the prototype is the cost of developing system design specifications for subsequent systems purchases, and the completion of one prototype tailored for a specific USAF "customer" as a means of transferring technology. The main development stream consists of the following recommended tasks, funding and schedule.

(All Funds Shown in $ Millions)	FY 73	74	75	76	77	78
1. Model Development	.15	.15				
2. Systems Studies	.200	.100	.05			
3. Security Kernel Design	.10	.15	.10			
4. Prototype Development		1.0	1.15	.5	.2	.2
ADP Support	.25	1.0	1.0	1.0	.5	.2
Totals	.70	2.40	2.30	1.5	.7	.4

4.2 Supporting Engineering Developments

The approach for the Supporting Engineering Development (Figure 4) has two components: the development of secure peripherals for use in resource-sharing systems and a handbook of computer security techniques.

There are several items of secure communications equipment whose development is required to enhance the security and reduce the cost of large-scale systems processing classified information. Two items requiring development are a low cost office environment (non-ruggedized) secure terminal (target cost: $3000), and a multiplexed crypto concentrator for use at central computer sites serving a large number of remote terminals. The objective for the office environment secure terminal is to provide a low cost integrated crypto device, container, and terminal that can be operated by personnel without crypto clearances, and will not require crypto vaulting. The lack of availability of such terminals severely limits the system design options presently available because of the very high cost of physically securing crypto equipment and its connections to remote terminals. It is important to recognize that this development is not dependent on the advanced development programs; rather it is needed even now to reduce costs associated with developing large scale information systems processing classified material. The objective for the multiplexed crypto concentrator is to significantly reduce the hardware,

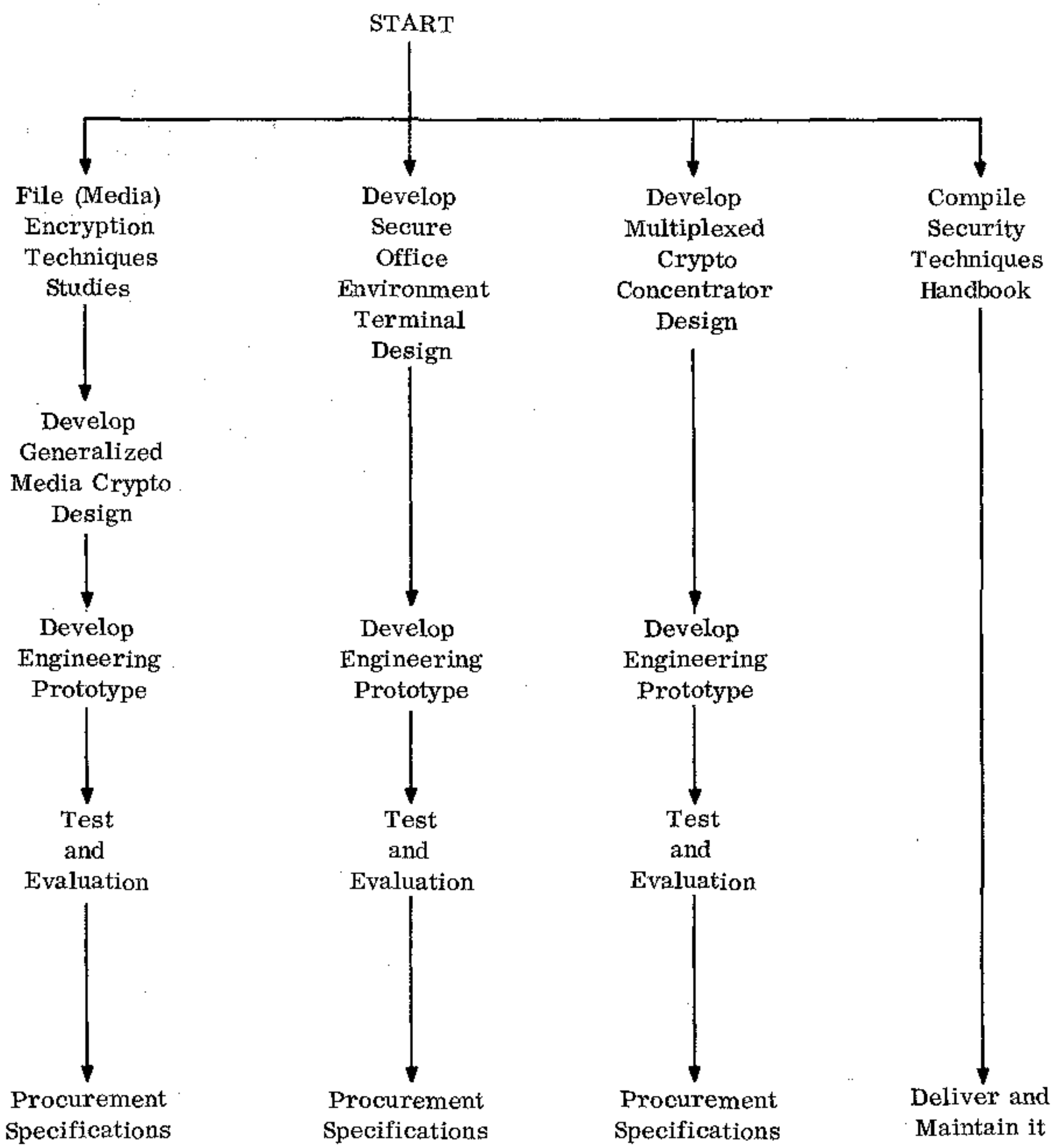


Figure 4. Supporting Engineering Development Approach

maintenance and operating costs of interfacing a large number of terminals to a central computer site by using a single crypto device for all terminals. In addition to the direct reduction of costs by not requiring one crypto device per line at the central site, there are added benefits of reduced space, power, air-conditioning, and maintenance possible with this approach.

Included with these developments is the use of file encryption techniques to alleviate the major existing problems of physically protecting magnetic media (tapes, discs, drums) containing classified information. Aside from the direct benefits possible by not requiring special handling of media containing classified information, this development will have significant benefits in tactical operations, or in any other environment where the risk of file media being captured or lost is high.

The handbook of computer security techniques is envisioned as a collection of system design, implementation, and operation practices covering all aspects of computer security from techniques of user identification through methods of program validation to recommended security policy, practices and procedures in the operation of secure systems. It is intended for use by designers and developers of USAF information systems. Because of anticipated changes in this technology, the handbook should be maintained throughout the indefinite future.

The funding and recommended schedule for these tasks are:

(All Funds Shown in $ Millions)	FY					
	73	74	75	76	77	78
1. Office Environment Secure Terminal	.1	1.45	.9	.2		
2. Multiplexed Crypto Concentrator	.2	.2	.3	.4	.1	
3. File (Media) Encryption Development	.15	.5	.35	.2		
4. Handbook of Security Techniques	.15	.1	.1	.1	.1	.1
Totals	.6	2.25	1.65	.9	.2	.1

4.3 Related Advanced Development Plan — Developments for Interim Solutions to Current Problems

After developing the secure computing model outlined above, it is recommended that the results be applied to current problems even on an interim basis. The two problem areas that can be affected are the implementation of secure limited use systems, and the evaluation of the feasibility of repairing current systems in the USAF inventory.

4. 3. 1 Secure Transaction Oriented Systems

The objective of this development item (shown in Figure 5) is to apply near-term results of the modeling activity to current USAF operational problems. It was noted by the requirements working group that many of the planned Air Force systems were transaction processing systems built around a data base management system (DMS) with a query language capability.

The degree of threat posed by a malicious user in this kind of environment is a function of the amount of programming he can do. For example, if the malicious user can only (legitimately) use an on-line transaction-oriented Query and DMS, his capability to affect the operation of the system is limited by the intrinsic capability of the tools he can use. Most transaction-oriented systems do not provide the malicious user with sufficient tools to take over control of the system; he cannot attack the system with his own programs. He may be able to gain unauthorized access to classified data by exploiting a pre-programmed weakness due to careless design or implementation, or planted as a 'trap door' in the application or in the programming and operating systems supporting the application. The security threat posed by this mode of use depends on whether the application is designed in such a way as to assume that each user is fully controlled in all actions he may take on the system. In addition both the application and the <u>programming and operating system</u> for the hardware supporting the application must be implemented by trustworthy (cleared) personnel in order to preclude the possible inclusion of 'trap doors'.

Because transaction-oriented systems are so prevalent in USAF applications, we recommend that the model be used as the base for developing a secure multilevel data management and query system as an interim way to obtain secure multilevel transaction systems. It appears feasible to augment the existing hardware and software controls in contemporary systems with a programmed reference validation interpreter, subject to the risk that trap doors have been inserted in the application or the software for the base machine. It may also be possible to use the same technique to support the general use of one or more of the higher order programming languages (only).

While any realistic assessment of the trap door threat would have to conclude that to date there is no evidence of malicious placement of trap doors in contemporary system software, there is no technical problem to doing so. Under present modes of operation where installations accept operating system updates and even whole revisions of an operating system without question, there is little doubt that the targeted system(s), could be induced to accept and install a trap door modification to their operating system.

Further, as long as present day commercial computer hardware is used to base even transaction-oriented systems, the complexity and size of the operating system programs running in supervisory (control) state leaves the practicability of analyzing them (or their revision) for trap doors in doubt.

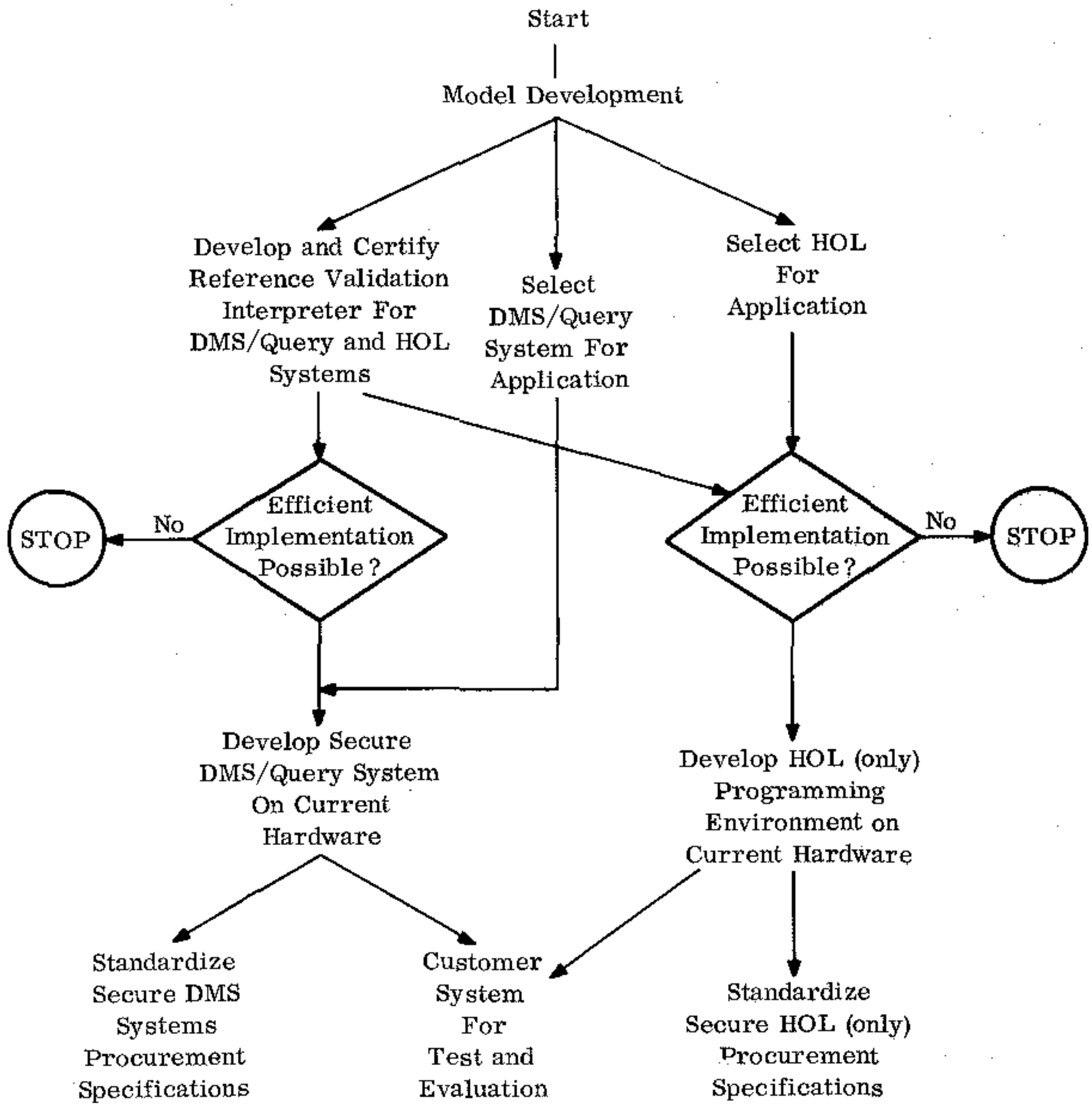


Figure 5. Interim Solutions to Current Problems — DMS/Query/HOL Systems

Because of the trap door threat, we would only recommend this development for those systems where no update or revision of the manufacturers software is required after the application is developed. This limitation is necessary because such systems will become attractive targets and trap doors can easily be added to the software maintained by the manufacturer.

The following tasks, funding and schedule are recommended:

(All Funds Shown in $ Millions)	FY					
	73	74	75	76	77	78
1. Develop Reference Validation Interpreter Design	.1	.2				
2. Implement DMS/Query System for a current system using (1) above	.2	.3	.2			
3. Develop Higher Order Language (only) programming environment on current systems using (1) above	.1	.2	.1			
Totals	.4	.7	.3			

4. 3. 2 <u>Repair of Current Systems</u>

The secure computing model can provide a basis for examining the design and implementation of contemporary computing systems and assessing the degree of effort required for their repair. The objective of this effort (shown in Figure 6) is to survey key contemporary systems to determine whether it is economically feasible to redesign and/or reimplement their operating systems to provide secure computing environments to the applications based on these systems.

The panel cannot overemphasize its belief that "patching" of known faults in the design or implementation of existing systems without any better technical foundation than is presently available, is futile for achieving multilevel security. We wish to distinguish, however, between the patching problem and the possibility of selective re-implementation of portions of an operating system to eliminate known security deficiencies and to provide a better technical foundation for the development of more secure systems for some environments. We do not see any method to provide the level of security desired by the Air Force for many of its systems through any simple technique or simple fix. It is also evident that re-implementation of nearly all contemporary systems would be necessary in order to provide even the minimum level of privacy necessary to implement need-to-know controls in all applications involving classified information. It is recommended that only those systems in widespread use be considered. Obviously, a prime candidate for such a system would be the WWMCCS using the Honeywell 6000 series equipment.

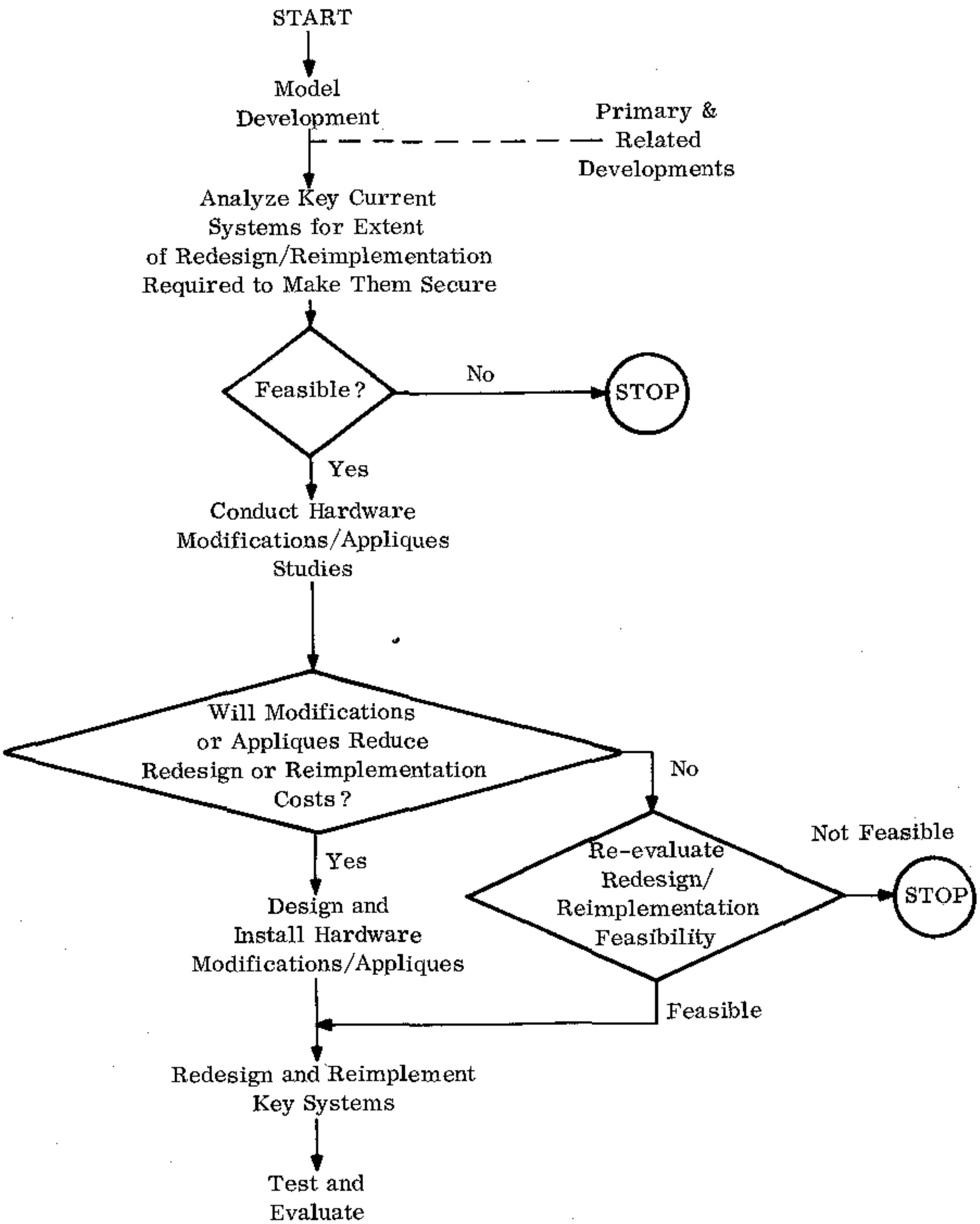


Figure 6. Approach to Repair of Current Systems

The feasibility of this approach is unknown; as a consequence, the plan starts with an analysis of the key system(s) to determine whether or not they can be altered to conform to the principles of the model. As a result of the analysis, the feasibility of the approach must be evaluated before continuing. Should such a course not be possible, then the only alternative is to severely restrict the type of operations on these systems until the results of the other technical developments are available for application to these systems. The tasks, funding, and schedule for this alternative plan for one system are shown below.

(All Funds Shown in $ Millions)	FY					
	73	74	75	76	77	78
1. Analyze key current system for extent of redesign or reimplementation	.1	.2	.2			
2. Conduct hardware modification/applique studies		.2	.2			
Subtotal for feasibility investigation (1 system)	.1	.4	.4			

Assuming that the analyses indicate feasibility of repairing or reimplementing a system, the additional tasks are:

	73	74	75	76	77	78
3. Design and Install Hardware Modifications or Appliques			.3	.5	.2	
4. Redesign and Reimplement Key System			.4	.8	.4	
Subtotals for Reimplementation	—	—	.7	1.3	.6	
Totals (1 system)	.1	.4	1.1	1.3	.6	

Detailed projections for subsequent systems are not shown, but are estimated to be 80-90% of the effort shown above for each additional system.

4.4 Exploratory Development Plan

The computer security exploratory development plan is a continuing effort complementing the advanced development and engineering plans, and is directed to exploring various alternatives to those that appear most attractive at present, and to developing additional methodology and techniques applicable to the problem of providing secure computing environments. Many of the items in the exploratory development plan appear in the advanced development plan as well, reflecting the fact that while our knowledge is sufficient to apply to the problems perceived today, both the knowledge and the perception of the problem will undergo significant change in the mid- to longer-term future. A properly integrated exploratory development program

can be used to provide continued guidance to the advanced development plan as well as to undertake work in anticipation of changes in requirements and technology. In some ways, the present 'crisis' over computer security is due to the fact that such an exploratory development program has not been part of the USAF's efforts in the past.

There are two main components to the exploratory development. The hardware techniques and architecture studies cover such topics as internal encryption and other techniques to eliminate the effects of inadvertent disclosure, direct execution higher order language machines, storage media techniques without magnetic residue and a computer aided integrated computer systems design environment.

System technology studies include automatic and derived classification techniques, data structure recovery techniques, security surveillance techniques, network studies and the like.

The tasks, funding and schedule for the Exploratory Development Plan are:

(All Funds Shown in $ Millions)	FY					
	73	74	75	76	77	78
1. Hardware & Architecture Studies	.45	.7	.7	.75	.45	.2
2. Systems Technology	1.15	1.95	1.95	1.05	.85	.75
Totals	1.60	2.65	2.65	1.80	1.30	.95

4.5 Summary Of The Development Plan

The advanced development plan for a secure prototype system and the supporting engineering development are shown graphically in Figure 7. The outputs of both of these developments include prototype hardware. A cost summary of the major items is shown below.

Cost Summary For Recommended Computer Security Program(s)
(All Amounts Shown in $ Millions)

	Fiscal Year						
	73	74	75	76	77	78	
I. Development of Secure Open-Use System Prototype							
1. Develop Model of Secure Resource Sharing	.15	.15					.30
2. Develop Security Kernel Design	.1	.15	.1				.35
3. Systems Studies	.2	.1	.05				.35
4. Prototype Development (includes ADP Support)	.25	2.0	2.15	1.5	.7	.4	7.0
TOTALS	.70	2.4	2.3	1.5	.7	.4	8.00

II. Supporting Engineering Developments							
1. Handbook of Computer Security Techniques	.15	.1	.1	.1	.1	.1	.65
2. Secure Office Environment Terminal	.1	1.45	.9	.2			2.65
3. Multiplexed Crypto Concentrator	.2	.2	.3	.4	.1		1.20
4. File Encryption Techniques	.15	.5	.35	.2			1.20
TOTALS	.60	2.25	1.65	.90	.20	.10	5.70

Because they are outside the main development stream, a related advanced development to provide interim solutions to current problems, and an exploratory development program in computer security are shown separately in Figure 8. Since the interim solutions development is addressing current problems, the funding for these items should come from existing programs. The figures shown are our estimate of what the effort will cost. The exploratory development program is directed to provide a continued influx of techniques and technology bearing on the problem of secure computing systems.

<u>Cost Summary for Related Developments and Exploratory Development Program</u>

(All Amounts Shown in $ Millions)

Fiscal Year

III. Developments for Interim Solutions to Current Problems	73	74	75	76	77	78	
1. Secure DMS/Query Systems	.4	.7	.3				1.4
2. Repair <u>One</u> Current System	.1	.4	1.1	1.3	.6		3.5
TOTALS	.5	1.1	1.4	1.3	.6		4.9

Fiscal Year

IV. Exploratory Development Plan	73	74	75	76	77	78	
1. Hardware Architectural Studies	.45	.70	.70	.75	.45	.20	3.25
2. Systems Technology	1.15	1.95	1.95	1.05	.85	.75	7.70
TOTALS	1.60	2.65	2.65	1.80	1.30	.95	10.95

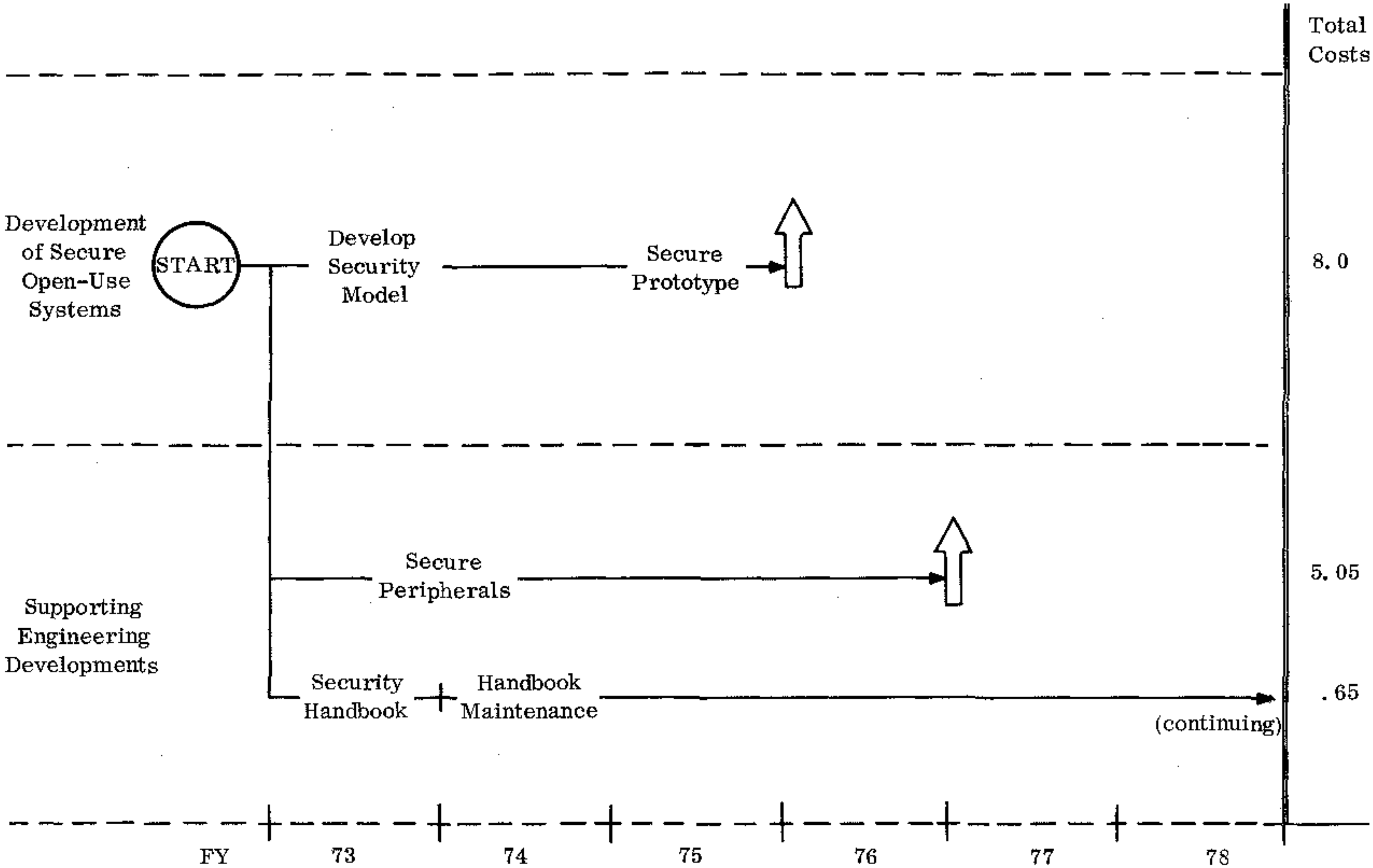


Figure 7. Development Plan

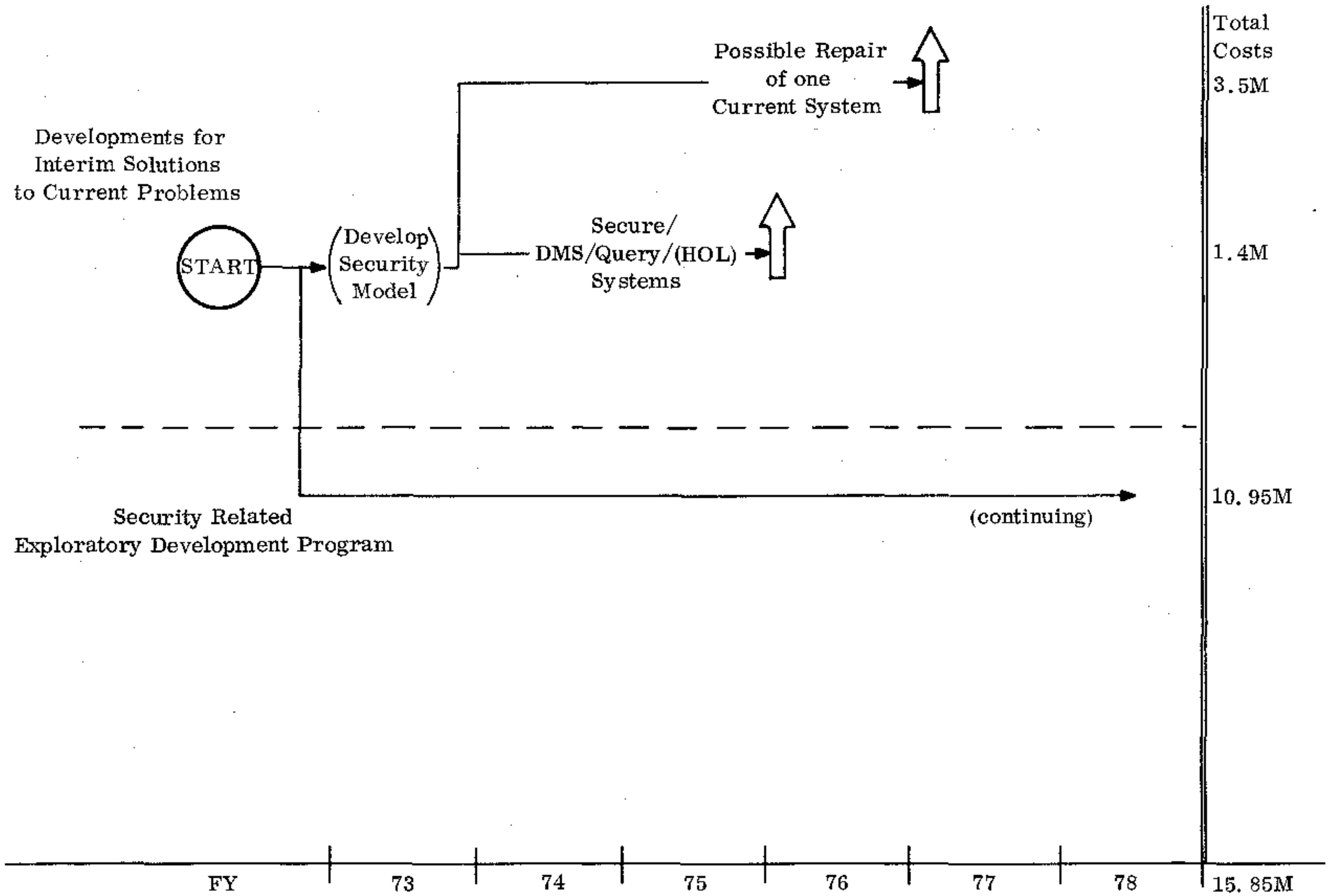


Figure 8. Interim Solutions and Research Plan

SECTION V

ECONOMIC CONSEQUENCES OF INADEQUACIES OF CURRENT SYSTEMS

The consequences of the inadequate protection mechanisms in current Air Force computer systems are both the potential for loss of information critical to national security by enemy penetration, and a higher cost of operation. The importance of timely and accurate information to effective military operations leads to taking of risks of penetration in order to achieve needed capabilities in command and control, intelligence, and logistic systems. This could result both in a continual loss of information and ultimately in a "big catastrophe" jeopardizing national security or provoking armed conflict.

In order to reduce these risks, many systems are operated far below their potential. This increases operational costs by:

- inefficient utilization of existing hardware and personnel, or the acquistion of extra people and machines in order to maintain separation of classified from unclassified information or to provide simultaneous processing of two or more levels of classified information
- loss of information accuracy, timeliness, and completeness resulting from increased processing time, reduced data sharing, and inadequate data correlation, brought about by having to maintain separation of classified from unclassified data or of two or more levels of classified information.

It is estimated that these costs amount to about $100, 000, 000 per year. [7]

5. 1 Cost of Inadequate Systems

In order to reduce the risk of enemy penetration of current systems, many of them are operated at levels of effectiveness far below their potential. Specifically, the following operational approaches are often found:

- separate computers for separate applications are used to achieve isolation, where combined operation on a single machine would significantly reduce costs;

---

[7] USAF Data Processing usage summary reported in Data MITRE Technical Report 2310, indicates annual costs for personnel and ADP equipment of approximately $342 million per year. We added $5 million/year for costs associated with creating and maintaining a secure physical environment for a total of $347 million. Using our estimate that 40% of these costs are due to the factors cited in 5. 1 yields on annual cost of classified processing of $100, 000, 000/year.

- sharing of a computer by two or more applications which could be run concurrently is accomplished by scheduling their use at different times with costly changeover and sanitization procedures;
- scheduling of classified computing at times, usually at night, when time-sharing terminals can be disconnected;
- restricting capabilities available through remote terminals;
- use of expensive crypto-devices and secure environments for each remote terminal, even those on which no classified processing takes place;
- clearing all personnel to the highest level of classified information processed by a system.

These operational practices require substantially more equipment and more personnel than would be required if the applications were performed on secure resource-sharing systems. The increase in cost per system due to the factors cited above ranges between 10% to over 100%, with a conservative estimate of the average increased cost per system to be around 40%.

In addition to the higher cost of computer operations, the reduced capabilities provided by current systems relative to those which could be provided by secure multilevel resource-sharing systems increase the cost of Air Force command and control, intelligence, and logistic systems by requiring manual handling of information that could be automated and by the loss of operational effectiveness resulting from information that is less timely, accurate, and complete than it should be.

5.2 Cost to Penetrate Systems

Another way to evaluate the cost effectiveness of the advanced development program is to consider what effect the plan for obtaining a secure prototype would have on the estimated costs to a potential enemy of penetrating systems. We can contrast the cost of the advanced development to obtain a penetration-proof system, with the cost of penetrating existing systems based on present technology. We can distinguish at least four cases of interest:

a) Contemporary systems 'As Is'.

b) Contemporary systems with known flaws repaired.

c) Selective reimplementation of contemporary systems.

d) Re-design and Re-implementation.

Case a) 'As Is'.

In this case there are essentially no costs for finding at least one exploitable vulnerability and then designing an attack around it. As was noted in previous sections, present systems are not designed with security requirements in mind. Based on

current experience with penetration exercises, and assuming the availability of an individual with technical familiarity with the target system, the cost to find and exploit at least one design or implementation flaw in virtually any contemporary system is one man-month of effort or less, and less than $1000 worth of computer time. The total costs are estimated at less than $3000.

Case b) 'Repaired' Contemporary Systems.

In this case, we consider the known operating system design and implementation flaws are repaired, but that the system is not re-designed and reimplemented. The primary difference between this case and the previous one is the cost of finding a residual exploitable vulnerability under the assumption that all previously known vulnerabilities are repaired, and that the repairs do not introduce new flaws. Because there is no systematic way of analyzing and testing an operating system for security without having a basis for knowing when a secure system has been achieved, it is estimated that at least one previously undetected vulnerability remains and that it can be found in three man-months or less. The balance of the attack is as in case (a). Assuming analyst costs of $2000/month and total machine time costs at $2000, the total cost of this attack is estimated to be $8000.

As an interesting sidelight, a large scale contemporary system was recently 'repaired' after a previously successful penetration exercise. The repairs, involving over 250 changes to the operating system, and subsequent testing by both the user and the manufacturer involved, took on the order of 10-15 man years of effort over a 6 month period. The cost of this effort is estimated at 5% of the proposed plan to obtain a secure prototype. A second penetration exercise against the 'repaired' system was successful in less than one man-week of effort.

Case c) Selective Reimplementation.

The feasibility of selective reimplementation of portions of contemporary operating systems was recommended for investigation as part of a related development plan directed to current problems. Assuming it is feasible for some system(s), the cost of selective reimplementation of parts of a system is estimated at 1.5-2.5 million per system (17%-28% of the cost of the plan to obtain a secure prototype).

There is no guarantee of success, and virtually no way to preclude intentional 'trap doors' unless the work is done under strict government control, and the unaffected parts of the system are unaltered.

Because selective reimplementation can only block the more obvious attack routes, and thus strengthen need to know controls, there are still possible attack avenues open to a malicious user, although with a lower probability of success or increased risk of detection. It is estimated that the cost of penetrating such a system is between $100,000 and $200,000.

Case d) Redesign and Reimplemention.

This case is the ultimate in repairs. In effect it starts all over again from scratch. Even with a good model of secure computing, the cost of redesign and reimplementation is likely to exceed $10 million dollars per system type. The costs for these various cases are shown in the Table below.

Case	Estimated Cost to Obtain (Per System)	% of Development Plan for Service Prototype	Estimated Cost to Penetrate by Malicious User
a) 'As Is' Systems	0	0	$3000
b) Repaired Systems	$500, 000	5%	$8000
c) Selective Reimplementation	1. 5-2. 5 million	17%-28%	$100, 000-$250, 000
d) Redesign & Reimplementation	$10 million	110%	blocked

SECTION VI

CONCLUSIONS

The panel concludes that there is an urgent requirement to develop secure resource-sharing systems for Air Force applications. The principle threat against which current systems provide no protection is a malicious user, although other threats are also present in most environments.

The present modes in systems processing classified information are unacceptable from a cost viewpoint and reduce Air Force operational effectiveness, due to archaic information handling procedures imposed for security reasons alone. Even the anticipated effect of WWMCCS is blunted because of severely restricted operating modes arising from security reasons.

In spite of the desire of the Air Force and other military departments to have security, merely saying a system is secure will not alter the fact that unless the security for a system is designed in at its inception, there are no simple measures to later make it secure.

If we had the results of the program, the following things would be possible where appropriate:

- A single system capable of simultaneously processing information of any classification for users of any clearance status with no risk that any actions on the part of any user will result in his obtaining any (classified) information not specifically authorized to him.
- Systems that will permit unclassified processing over unprotected communications lines simultaneously with classified processing over protected lines, with no risk that external penetration through the unprotected communications will result in access to classified information.
- Systems that provide full programming capability to all users regardless of clearance and regardless of the maximum or instant highest classification of data being processed by the system.
- Systems that can run 'system programs' produced by anyone with no risk that an implanted or residual 'trap door' can be activated by any user on whose behalf the program is run to access (classified) data not specifically authorized to him.
- The ability to operate a terminal for classified processing without having to build a separate vault for storing the terminal crypto equipment when it is not being used.

- Free interconnection of systems into networks with no risk that any user of a network node can gain access to the network or any of its data if he is not specifically authorized to do so.
- The ability to allocate computer resources on the basis of need alone, with no regard for the classification of the data processed.

Finally, if nothing is done by the Air Force in this area, there is little hope that spontaneous efforts will provide the technology needed. The situation will become even more acute in the future as potential enemies recognize the attractiveness of Air Force data systems as intelligence targets, and perceive how little effort is needed to subvert them.

UNCLASSIFIED

Security Classification

DOCUMENT CONTROL DATA - R & D

(Security classification of title, body of abstract and indexing annotation must be entered when the overall report is classified)

1. ORIGINATING ACTIVITY (Corporate author) JAMES P. ANDERSON COMPANY BOX 42 FORT WASHINGTON, PENNA. 19034	2a. REPORT SECURITY CLASSIFICATION UNCLASSIFIED 2b. GROUP

3. REPORT TITLE

COMPUTER SECURITY TECHNOLOGY PLANNING STUDY

4. DESCRIPTIVE NOTES (Type of report and inclusive dates)

FINAL REPORT FEB. – SEPT. 1972

5. AUTHOR(S) (First name, middle initial, last name)

JAMES P. ANDERSON

6. REPORT DATE	7a. TOTAL NO. OF PAGES	7b. NO. OF REFS
OCTOBER, 1972	40	

8a. CONTRACT OR GRANT NO. F19628-72-C-0198	9a. ORIGINATOR'S REPORT NUMBER(S)
b. PROJECT NO.	
c.	9b. OTHER REPORT NO(S) (Any other numbers that may be assigned this report)
d.	ESD-TR-73-51, Vol. I

10. DISTRIBUTION STATEMENT

Approved for public release; distribution unlimited.

11. SUPPLEMENTARY NOTES	12. SPONSORING MILITARY ACTIVITY
	ELECTRONIC SYSTEMS DIVISION, AFSC L. G. HANSCOM FIELD BEDFORD, MASS. 01730

13. ABSTRACT

THE RESULTS OF A PLANNING STUDY FOR USAF COMPUTER SECURITY REQUIREMENTS ARE PRESENTED. THE STUDY RECOMMENDS RESEARCH AND DEVELOPMENT URGENTLY NEEDED TO PROVIDE SECURE INFORMATION PROCESSING SYSTEMS FOR COMMAND AND CONTROL AND SUPPORT SYSTEMS FOR THE AIR FORCE.

14. KEY WORDS	LINK A		LINK B		LINK C	
	ROLE	WT	ROLE	WT	ROLE	WT
COMPUTER SECURITY RESEARCH AND DEVELOPMENT SECURITY MODELS						